Official 6/19/03

REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 1, 3-5 and 9 are amended. Claims 1-25 remain pending in this application.

35 U.S.C. § 103

Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,035,121 issued to Chiu et al. (hereinafter "Chiu") in view of U.S. Patent No. 5,634,114 issued to Shipley (hereinafter "Shipley"). Applicant respectfully traverses the rejection.

Chiu describes systems and methods for converting a computer program component from a first language to a second language. The techniques utilize a "leverage tool" (i.e. the application program referred to in the Office Action) that analyzes three components, or DLLs, to-wit: (1) a current version of a component in the first language; (2) a previous version of the component in the first language; and (3) a previous version of the component in the second language. The leverage tool creates a new component, or DLL, that is a current version of the component in the second language.

Shipley discloses systems and methods for DLL version negotiation, which consists of an application specifying a particular version of a DLL when the application calls the DLL during application execution. The DLL refers to an internal table that specifies version numbers supported by the DLL. If the specified version is supported, processing continues normally. Otherwise, the DLL returns an error flag to the application together with a list of versions that are supported. The application refers to its own table to determine if any of the

supported versions are compatible with the application. If so, processing continues. If not, the application performs an error trap.

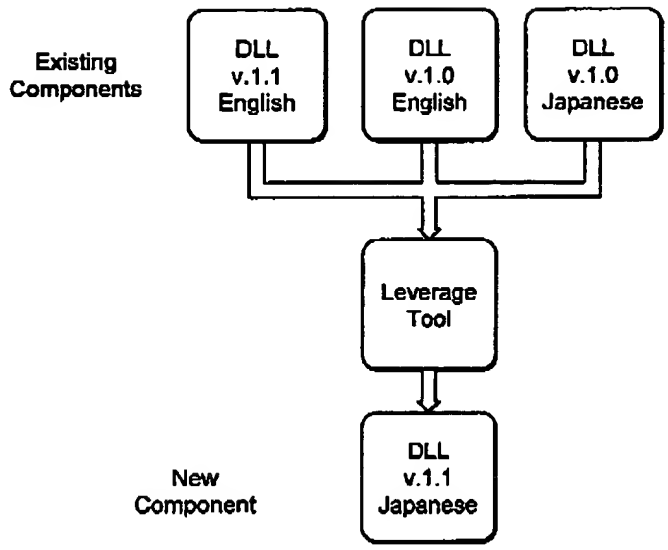
Claim 1 is amended to recite a method comprising the steps of "storing a computer application program on one or more computer-readable media" and "storing a first version of a shared component in the one or more computer-readable media for execution on a computer system that stores at least a second version of the shared component, **wherein the first component is a functional component of the computer application program that is compatible therewith.**" (Amendment emphasized). A logical relationship is established "between the computer application program and the first version of the shared component so that the application uses the first version of the shared component **and not the second version of the shared component** when the application is executed on the computer system." (Amendment emphasized).

The first version of the shared component of claim 1 is a program that provides executable functionality to the application program ("storing a first version . . . for execution . . . ."). In other words, the first version of the shared component must be executable. Furthermore, the first version of the shared component is specifically a component of the application program and it is compatible with the application program.

Chiu does not teach or suggest this type of shared component. The shared components described by Chiu are not components of the application program.

The following figure generally depicts the functionality of the systems and methods described in Chiu. For the following example, a current version of a component is designated as v.1.1, and a previous version of the component is

designated as v.1.0. The first language is "English" and the second language is "Japanese."



The "application program" of Chiu, i.e. the leverage tool, does not receive executable functionality from any of the components cited in the Office Action. In other words, the components are not components of the leverage tool, but are components of one or more other programs. The leverage tool merely analyzes the contents of the shared components to derive and create a new component. As such, the shared components are not components as is required by claim 1.

Shipley does not teach or suggest storing different versions of a shared component (DLL) wherein at least one of the versions of the shared components are executed with an application program. Shipley teaches determining if a component stored on a system is compatible with an application that calls the component. The process described by Shipley is carried out even if there is only

one version of the component in the system. This is contrary to the recitations of claim 1.

Accordingly, claim 1 is allowable over the cited references and the rejection thereof should be withdrawn.

Claim 2 depends from claim 1 and is allowable at least by virtue of that dependency. Furthermore, claim 2 further defines "the establishing a logical relationship between the computer application program and the first version of the shared component." that is recited in claim 1. This establishing step further comprises "configuring a logical directory data structure that has multiple logical directories so that the computer application program and the first version of the shared component are referenced within a first logical directory, and wherein the second version of the shared component is referenced within a second logical directory."

In the rejection of claim 2, the Office Action states that Chiu "further teaches configuring a logical directory data structure that has multiple logical directories. . . ."

In the previous rejection (i.e. the rejection of claim 1), the Office Action states that "Chiu does not explicitly teach establishing a logical relationship between the computer application program and the first version of the shared component so that the application uses to [sic] the first version of the shared component when the application is executed on the computer system..." Office Action § 5 (page 3).

Therefore, the language explaining the rejection of claim 2 is contradictory in that it states that Chiu teaches such an establishment of a logical relationship. Furthermore, the references for this language include Fig. 2 and reference

numerals 125-145 of Fig. 5. Applicant contends that these figures do not show – either implicitly or explicitly – any directory structure from which the teachings of claim 2 could be inferred.

For this additional reason, claim 2 is allowable over the cited references and the rejection thereof should be withdrawn.

Claim 3 depends from claim 2 and is allowable at least by virtue of that dependency. Furthermore, the excerpt cited by the Office (col. 4, lines 55-63) does not discuss or imply the "storing a reference to an indicator in the logical directory where the computer application program and the first version of the shared resource are referenced, the indicator indicating to the computer application that the first version of the shared resource referenced by the indicator is referenced in the logical directory where the computer application is referenced" that is recited in claim 3. Neither does the reference to Fig. 3 discuss particular memory partitions (i.e. directories) where the computer application program or the shared component is stored.

For this additional reason, claim 3 is allowable over the cited references and the rejection thereof should be withdrawn.

Claim 4 depends from claim 1 and is allowable at least by virtue of the dependency. Accordingly, the rejection of claim 4 should be withdrawn.

Claim 5 recites a method comprising the steps of "calling a shared component in a computer system", "detecting a local file that indicates the presence of a locally-stored version of the shared component, **the local file being a different file than the shared component itself**" and "in response to detecting the local file, utilizing the locally-stored version of the shared component instead of a global version of the shared component present in the computer system."

The Office Action (p. 5, ¶2) states that element *145* of Fig. 5 in Chiu indicates that Chiu "teaches detecting a local file that indicates the presence of a locally-stored version of the shared component . . . and in response to detecting the local file, utilizing the locally-stored version of the shared component instead of a global version of the shared component present in the computer system."

Element *145* of Fig. 5 (Chiu) represents the final component that is derived by the leverage tool from the two older versions and one newer version of the component. The only way element *145* can have a relationship to the local file recited in claim 5 would be if element *145* was the local file. However, the amendment to claim 5 (emphasized above) clarifies that the local file is a separate file from the shared component. Therefore, Chiu as applied, does not teach or suggest the elements required by claim 5.

Accordingly, claim 5 is allowable over the cited references and the rejection thereof should be withdrawn.

Claims 6 - 8 depend from claim 5 and are allowable at least by virtue of that dependency. Therefore, the rejection of these claims should be withdrawn.

Claim 9 has been amended and now recites one or more computer-readable media containing computer-executable instructions that, when executed on the computer, perform the following steps: "storing a computer application program in a computer system" and "storing a first version of a shared component in the computer system for execution on the computer system, the computer system storing at least a second version of the shared component" and "**wherein the computer application program is configured to utilize the first version of the shared component and not the second version of the shared component when**

the computer application program is executed on the computer." (Amendment emphasized).

The amendment to claim 9 makes clear that only the first version of the shared component and not the second version of the shared component is used. Chiu analyzes two components and amends a third component to derive a fourth component. Clearly, Chiu does not use only one of the versions of the shared component.

Accordingly, claim 9 is allowable over the cited references and the rejection of claim 9 should be withdrawn.

Claims 10 - 13 depend from claim 9 and are allowable at least by virtue of that dependency. Accordingly, the rejection of these claims should be withdrawn.

Claim 14 recites a computer system that includes "memory divided into a plurality of discrete partitions," "a first application program stored in a first memory partition," "a first version of a shared component stored in a second memory partition, the first version of the shared component useable by at least a second application program" and "a second version of the shared component stored in the first memory partition." The computer system also includes "an indicator that, when present, indicates the existence of the second version of the shared component" and "wherein the first application utilizes the second version of the shared component if the indicator is present."

Neither Chiu nor Shipley teaches or suggests the elements recited in claim 14. Neither reference discusses using memory partitions as a part of a process of determining which of multiple shared components should be utilized by an application program.

Therefore, claim 14 is allowable over the cited references and the rejection thereof should be withdrawn.

Claims 15-21 depend from claim 9 and are allowable at least by virtue of that dependency. The rejection of these claims, therefore, should be withdrawn.

Claim 22 recites a directory tree data structure stored on one or more computer-readable media and having multiple directories. The directory tree data structure includes "a first directory that contains a pointer to a first version of a shared component useable by a plurality of computer programs," "a second directory that contains a pointer to an application program and a pointer to a second version of the shared component" and "wherein the application program utilizes the second version of the shared component when the application program calls the shared component."

Neither of the cited references teaches or suggest using a directory tree data structure as part of a process to determine which of several versions of a shared component should be utilized by an application program.

Accordingly, claim 22 is allowable over the cited references and the rejection of claim 22 should be withdrawn.

Claims 23-25 depend from claim 22 and are allowable at least by virtue of that dependency. Accordingly, the rejection of these claims should be withdrawn.

Official 6/19/03

Conclusion

All pending claims 1-25 are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the subject application. If any issues remain that prevent issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Date: 6/19/03

By: _____
James R. Banowsky
Reg. No. 37,773
(509) 324-9256x216